

cin.ufpe.br



UNIVERSIDADE FEDERAL DE PERNAMBUCO

# Medindo Desempenho dos Processadores

Edna Barros

# Agenda

---



- Entender conceito de desempenho.
- Métricas para medir desempenho
  - Tempo de execução
  - Taxa de processamento
- Tempo de Execução
- CPI
- Medindo o CPI
- Avaliando computadores



# O que é Desempenho?

Airplane	Passenger capacity	Cruising range (miles)	Cruising speed (m.p.h.)	Passenger throughput (passengers × m.p.h.)
Boeing 777	375	4630	610	228,750
Boeing 747	470	4150	610	286,700
BAC/Sud Concorde	132	4000	1350	178,200
Douglas DC-8-50	146	8720	544	79,424

FIGURE 1.14 The capacity, range, and speed for a number of commercial airplanes. The last column shows the rate at which the airplane transports passengers, which is the capacity times the cruising speed (ignoring range and takeoff and landing times).

# Response Time e Throughput



- **Tempo de Resposta**

- Tempo entre o início e a conclusão de uma tarefa, como observado pelo usuário final
- Tempo de resposta = Tempo de CPU + Tempo de espera (I / O, OS programação, etc)

# Response Time e Throughput



- **Taxa de Execução**

- Número de tarefas que a máquina pode executar em um determinado período de tempo
- Diminuindo o tempo de execução melhora o rendimento
  - Exemplo: usando uma versão mais rápida de um processador
  - Menos tempo para executar uma tarefa = mais tarefas podem ser executadas
- Aumentar a taxa de transferência também pode melhorar o tempo de resposta
  - Exemplo: aumento do número de processadores em um multiprocessador
  - Mais tarefas podem ser executadas em paralelo
  - O tempo de execução de tarefas individuais sequencial não é alterado
  - Mas menos tempo de espera em fila de agendamento reduz o tempo de resposta

# Definição de Desempenho

- Para algum programa executando na máquina  $X$

$$\text{Desempenho}_X = \frac{1}{\text{TempoExecução}_X}$$

- $X$  é  $n$  vezes mais rápida que  $Y$

$$\frac{\text{Desempenho}_X}{\text{Desempenho}_Y} = \frac{\text{TempoExecução}_Y}{\text{TempoExecução}_X} = n$$

# Tempo de Execução



- Tempo de resposta

- Contagem corrida do tempo:
  - Esperando o tempo de entrada / saída, acesso ao disco, agendamento de OS, etc ...
- Número Útil , mas muitas vezes não é bom para fins de comparação

- Foco: CPU Execution Time

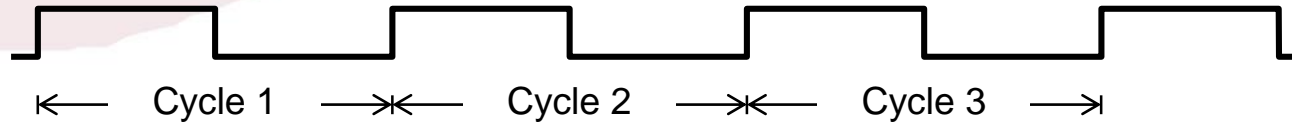
- Tempo gasto ao executar as instruções do programa
- Não conta o tempo de espera para escalonamento de E/S ou OS
- Pode ser medido em segundos, ou
- Pode estar relacionado ao número de ciclos de clock da CPU



# Ciclos de Clock



- Clock cycle = Clock period =  $1 / \text{Clock rate}$



- Clock rate = Clock frequency = Cycles per second

- 1 Hz = 1 cycle/sec
- 1 MHz =  $10^6$  cycles/sec
- 2 GHz clock has a cycle time =  $1/(2 \times 10^9) = 0.5$  nanosecond (ns)
- 1 KHz =  $10^3$  cycles/sec
- 1 GHz =  $10^9$  cycles/sec

Nós usamos frequentemente ciclos de clock para medir o tempo de execução da CPU

$$\text{CPU Tempo Execução} = \text{CPU ciclos} \times \text{tempoCiclo} = \frac{\text{CPU ciclos}}{\text{Clock rate}}$$



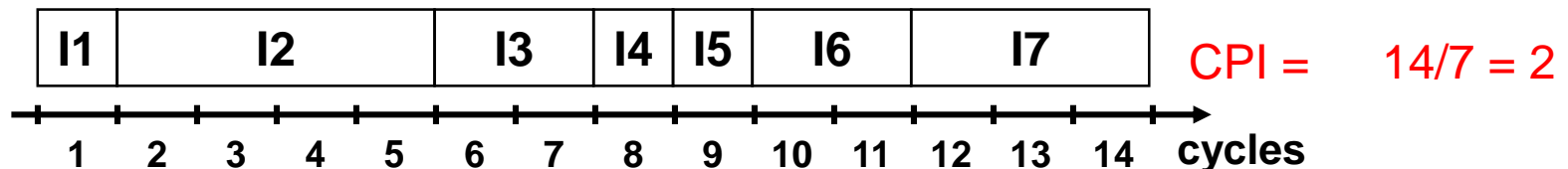
# Melhorando o Desempenho



- Para melhorar o desempenho, nós precisamos
  - Reduzir o número de ciclos de clock exigido pelo programa, ou
  - Reduzir o tempo de ciclo de clock (aumentar a taxa de clock)
- Exemplo:
  - Um programa é executado em 10 segundos no computador X com 4 GHz de clock
  - Queremos o projeto do computador Y para executar o mesmo programa em 6 segundos. Sendo que o computador Y pode exigir 20% mais ciclos para executar programa, qual deve ser o clock do computador Y?
  - Qual é o número de ciclos de CPU no computador X?
  - Qual é a velocidade do clock para o computador Y?
- Solução:
  - Ciclos no computador  $X = 10 \text{ sec} \times 4 \times 10^9 \text{ cycles/s} = 40 \times 10^9$
  - Ciclos no computador  $Y = 1.2 \times 40 \times 10^9 = 48 \times 10^9 \text{ cycles}$
  - Clock rate para computador  $Y = 48 \times 10^9 \text{ cycles} / 6 \text{ sec} = 8 \text{ GHz}$

# Clock Cycles per Instruction (CPI)

- Instruções podem usar um número diferente de ciclos para executar
  - Multiplicação leva mais tempo do que a adição
  - Operações de ponto flutuante demoram mais que inteiro
  - Acessar a memória leva mais tempo do que o acesso aos registradores
- CPI é um número médio de ciclos de clock por instrução



- Ponto Importante:
  - Alterar o tempo do ciclo, muitas vezes requer alteração no número de ciclos necessários para várias instruções

# Equação do Desempenho



- A execução de um determinado programa vai exigir ...
  - número de instruções de máquina
  - número de ciclos de clock
  - número de segundos
- Podemos relacionar os ciclos da CPU relógio para contagem de instruções

$$\text{Ciclos CPU} = \text{Número Instruções} \times \text{CPI}$$

- Equação do Desempenho: (relacionado com a contagem de instruções)

$$\text{Tempo} = \text{Número Instruções} \times \text{CPI} \times \text{tempo ciclo}$$

# Entendendo Equação do Desempenho

$$\text{Tempo} = \text{Número Instruções} \times \text{CPI} \times \text{tempo ciclo}$$

	Nr. Instr.	CPI	Tempo Ciclo
Programa (Algoritmo e Linguagem)	X		
Compilador	X	X	
ISA	X	X	X
Organização		X	X
Tecnologia			X

# Usando Equação do Desempenho



- Suponha que temos duas implementações do mesmo ISA
- Para uma determinada aplicação
  - A máquina A (multi-ciclo) tem um tempo de ciclo do clock de 250 ps e um CPI médio de 2.0
  - Máquina B (mono-ciclo) tem um tempo de ciclo de clock de 800 ps e um CPI de 1.0
  - Qual máquina é mais rápida para esta aplicação, e de quanto?
- Solução:
  - Ambos computadores executam o mesmo número de instruções =  $I$
  - CPU execution time (A) =  $I \times 2.0 \times 250 \text{ ps} = 500 \times I \text{ ps}$
  - CPU execution time (B) =  $I \times 1.0 \times 800 \text{ ps} = 800 \times I \text{ ps}$
  - Computador A é mais rápido que computador B por um fator de  $1.6 = \frac{800 \times I}{500 \times I}$

# Determinando o CPI

- Diferentes tipos de instruções com CPI diferentes
  - Seja  $CPI_i$  = ciclos por instrução para Classe i de instruções
  - Seja  $C_i$  = número de instruções para a classe i de instruções

$$\text{Ciclos CPU} = \sum_{i=1}^n (CPI_i \times C_i)$$

$$CPI = \frac{\sum_{i=1}^n (CPI_i \times C_i)}{\sum_{i=1}^n C_i}$$

- Projetistas muitas vezes obtêm CPI por uma simulação detalhada
- Contadores de hardware são usados também para calcular CPIs

# Exemplo: cálculo do CPI



- Problema

- Um projetista de compilador está tentando decidir entre duas sequências de código para uma máquina específica. Com base na implementação de hardware, existem três classes diferentes de instruções: classe A, classe B e classe C, e exigem um, dois e três ciclos por instrução, respectivamente.
- A primeira sequência de código possui 5 instruções: 2 de A, 1 de B e 2 de C
- A segunda sequência possui 6 instruções: 4 de A, 1 de B e 1 de C

- Calcular os ciclos de CPU para cada sequência. Qual sequência é mais rápida?

- Qual é o CPI para cada sequência?

- Solução

CPU cycles (1ª. sequencia) =  $(2 \times 1) + (1 \times 2) + (2 \times 3) = 2 + 2 + 6 = 10$  ciclos

CPU cycles (2ª. sequencia) =  $(4 \times 1) + (1 \times 2) + (1 \times 3) = 4 + 2 + 3 = 9$  ciclos

Segunda sequencia é mais rápida mesmo executando uma instrução a mais:

CPI (1ª. sequencia) =  $10/5 = 2$

CPI (2ª. sequencia) =  $9/6 = 1.5$



# Segundo Exemplo CPI

• Dado um mix de instruções de um programa em um processador RISC:

- Qual é a CPI média?
- Qual é o percentual de tempo utilizado por cada classe de instrução?

•Class <sub>i</sub>	Freq <sub>i</sub>	CPI <sub>i</sub>	CPI <sub>i</sub> × Freq <sub>i</sub>	%Tempo
•ALU	50%	1	$0.5 \times 1 = 0.5$	$0.5/2.2 = 23\%$
•Load	20%	5	$0.2 \times 5 = 1.0$	$1.0/2.2 = 45\%$
•Store	10%	3	$0.1 \times 3 = 0.3$	$0.3/2.2 = 14\%$
•Branch	20%	2	$0.2 \times 2 = 0.4$	$0.4/2.2 = 18\%$

$$\text{CPI Médio} = 0.5 + 1.0 + 0.3 + 0.4 = 2.2$$

# Número Instruções como Métrica de Desempenho



- MIPS: Millions Instructions Per Second
- Algumas vezes usada como métrica de desempenho
  - Máquinas rápidas □ valor de MIPS alto
- MIPS especifica a taxa de execução das instruções

$$\text{MIPS} = \frac{\text{Nr. Instruções}}{\text{Tempo Execução} \times 10^6} = \frac{\text{Taxa Clock}}{\text{CPI} \times 10^6}$$

- Podemos relacionar tempo de execução a MIPS

$$\text{Tempo Execução} = \frac{\text{Nr. Inst.}}{\text{MIPS} \times 10^6} = \frac{\text{Nr. Inst.} \times \text{CPI}}{\text{Taxa Clock}}$$



# Problemas em usar taxa de execução (MIPS)

---



- Três problemas com a métrica MIPS representando uma métrica de desempenho
- Não leva em conta a capacidade de instruções
  - Não é possível usar o MIPS dos computadores para comparar conjuntos de instruções diferentes, porque a contagem de instruções será diferente
- MIPS varia entre os programas no mesmo computador
  - Um computador pode não ter uma classificação MIPS único para todos os programas
- MIPS pode variar inversamente com o desempenho
  - Uma classificação elevada do MIPS nem sempre significa melhor desempenho
  - Exemplo no próximo slide mostra esse comportamento anômalo



# MIPS exemplo



- Dois compiladores diferentes estão sendo testados no mesmo programa para uma máquina de 4 GHz com três classes diferentes de instruções: Classe A, Classe B e Classe C, que exigem 1, 2 e 3 ciclos, respectivamente.
- A contagem de instruções produzida pelo primeiro compilador é 5000 milhões de instruções da Classe A, 1 bilhão de instruções da Classe B, e 1 bilhão de instruções da classe C.
- O segundo compilador produz 10.000 milhões de instruções da Classe A, 1 bilhão de instruções da Classe B, e 1 bilhão de instruções da classe C.
- Qual compilador produz código com o MIPS superior?
- Qual compilador produz código com o melhor tempo de execução?

# Solução do MIPS exemplo



- Primeiro, encontramos os ciclos de CPU para ambos os compiladores
  - Ciclos CPU (compilador 1) =  $(5 \times 1 + 1 \times 2 + 1 \times 3) \times 10^9 = 10 \times 10^9$
  - Ciclos CPU (compilador 2) =  $(10 \times 1 + 1 \times 2 + 1 \times 3) \times 10^9 = 15 \times 10^9$
- Depois, calculamos o tempo de execução para ambos os compiladores
  - Tempo Execução (compilador 1) =  $10 \times 10^9 \text{ ciclos} / 4 \times 10^9 \text{ Hz} = 2.5 \text{ seg.}$
  - Tempo Execução (compilador 2) =  $15 \times 10^9 \text{ ciclos} / 4 \times 10^9 \text{ Hz} = 3.75 \text{ seg.}$
- **Compilador 1 gera programa mais rápido**
- Agora compute a taxa MIPS para ambos compiladores.
  - $\text{MIPS} = \text{Nr. Instruções} / (\text{Tempo Execução} \times 10^6)$
  - $\text{MIPS (compilador 1)} = (5+1+1) \times 10^9 / (2.5 \times 10^6) = 2800$
  - $\text{MIPS (compilador 2)} = (10+1+1) \times 10^9 / (3.75 \times 10^6) = 3200$
- **Assim, o código do compilador 2 tem uma taxa MIPS de execução mais elevada.**

# Benchmarks



- Desempenho é melhor avaliado usando aplicações reais
  - Uso de programas típicos da área de aplicação
  - Programa representativos da classe de aplicações.
  - Exemplos: compiladores, editores, aplicações científicas, gráficos, ...
- SPEC (System Performance Evaluation Corporation)
  - Fundado e suportado por fabricantes e vendedores de computadores.
  - Companhias definiram um conjunto REAL de aplicações
  - Vários benchmarks para ...
    - CPU performance, graphics, high-performance computing, client-server models, file systems, Web servers, etc.
  - Interessante indicador de desempenho (e tecnologia de compiladores)

# SPEC CPU2000 Benchmarks



12 Integer benchmarks (C and C++)		14 FP benchmarks (Fortran 77, 90, and C)	
Name	Description	Name	Description
gzip	Compression	wupwise	Quantum chromodynamics
vpr	FPGA placement and routing	swim	Shallow water model
gcc	GNU C compiler	mgrid	Multigrid solver in 3D potential field
mcf	Combinatorial optimization	applu	Partial differential equation
crafty	Chess program	mesa	Three-dimensional graphics library
parser	Word processing program	galgel	Computational fluid dynamics
eon	Computer visualization	art	Neural networks image recognition
perlbmk	Perl application	equake	Seismic wave propagation simulation
gap	Group theory, interpreter	facerec	Image recognition of faces
vortex	Object-oriented database	ammp	Computational chemistry
bzip2	Compression	lucas	Primality testing
twolf	Place and route simulator	fma3d	Crash simulation using finite elements
		sixtrack	High-energy nuclear physics
		apsi	Meteorology: pollutant distribution

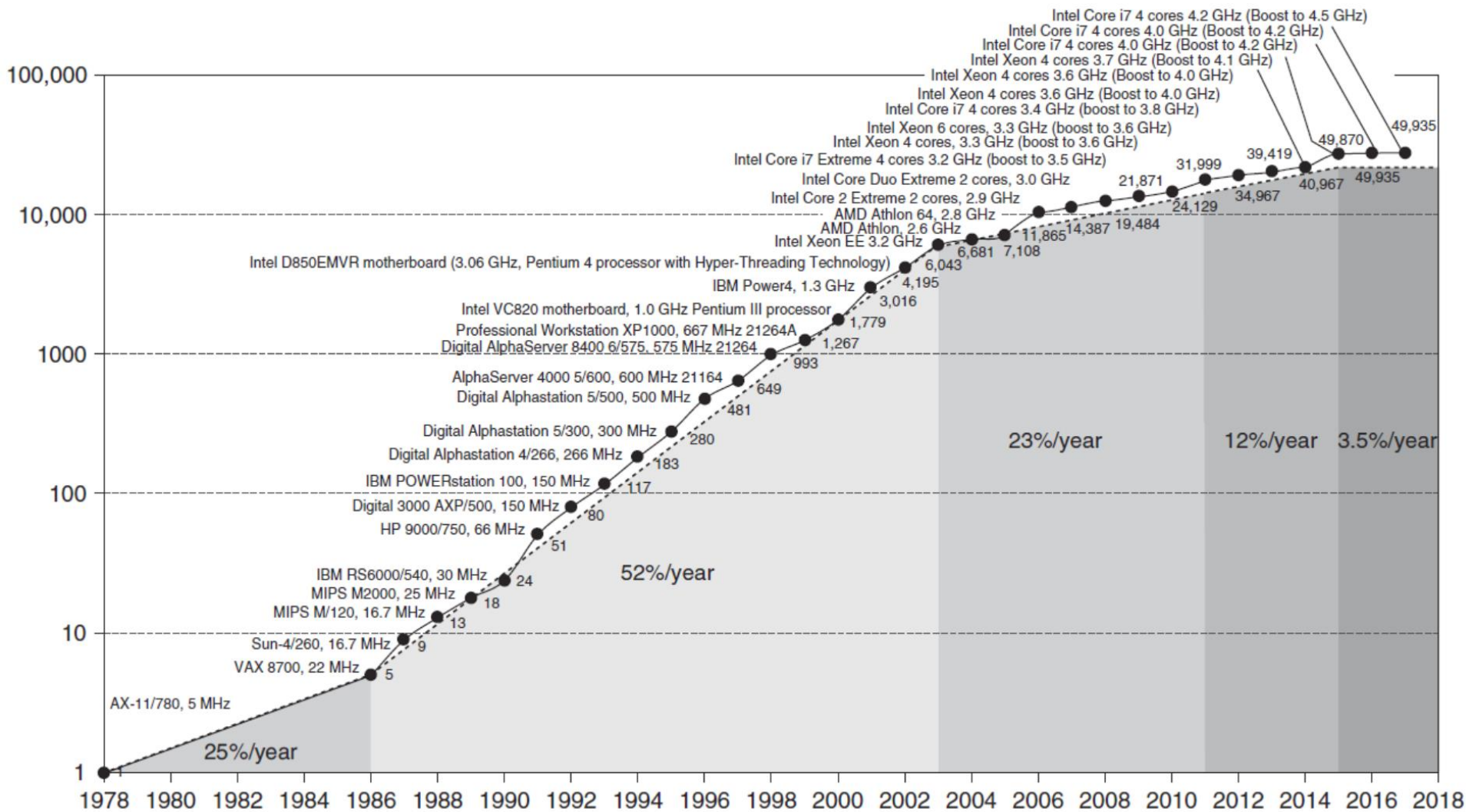
Taxa clock é usado como métrica

Benchmarks medem tempo de CPU devido ao pouco I/O





# Desempenho do Processador





# Desempenho processadores

Description	Name	Instruction Count x 10 <sup>9</sup>	CPI	Clock cycle time (seconds x 10 <sup>-9</sup> )	Execution Time (seconds)	Reference Time (seconds)	SPECratio
Interpreted string processing	perl	2252	0.60	0.376	508	9770	19.2
Block-sorting compression	bzip2	2390	0.70	0.376	629	9650	15.4
GNU C compiler	gcc	794	1.20	0.376	358	8050	22.5
Combinatorial optimization	mcf	221	2.66	0.376	221	9120	41.2
Go game (AI)	go	1274	1.10	0.376	527	10490	19.9
Search gene sequence	hmmer	2616	0.60	0.376	590	9330	15.8
Chess game (AI)	sjeng	1948	0.80	0.376	586	12100	20.7
Quantum computer simulation	libquantum	659	0.44	0.376	109	20720	190.0
Video compression	h264avc	3793	0.50	0.376	713	22130	31.0
Discrete event simulation library	omnetpp	367	2.10	0.376	290	6250	21.5
Games/path finding	astar	1250	1.00	0.376	470	7020	14.9
XML parsing	xalancbmk	1045	0.70	0.376	275	6900	25.1
Geometric mean	—	—	—	—	—	—	25.7

FIGURE 1.18 SPECINTC2006 benchmarks running on a 2.66 GHz Intel Core i7 920. As the equation on page 35 explains, execution time is the product of the three factors in this table: instruction count in billions, clocks per instruction (CPI), and clock cycle time in nanoseconds. SPECratio is simply the reference time, which is supplied by SPEC, divided by the measured execution time. The single number quoted as SPECINTC2006 is the geometric mean of the SPECratios.

# Desempenho e Potência



- Potência é um fator limitante
  - Capacidade das baterias melhora pouco
- Necessidade de projeto de processadores eficientes
- Redução de potência por:
  - Redução da frequência
  - Redução da Tensão
  - Colocando componentes para “dormir”
- Eficiência energética
  - Métrica importante para aplicações limitadas por consumo de energia
  - Definida como o desempenho dividido pela potência

- Ciclo: Energia = carga capacitiva x tensão<sup>2</sup>
- Transição: Energia = ½ carga capacitiva x tensão<sup>2</sup>
- Potência = ½ carga capacitiva x tensão<sup>2</sup> x freq. Chav.

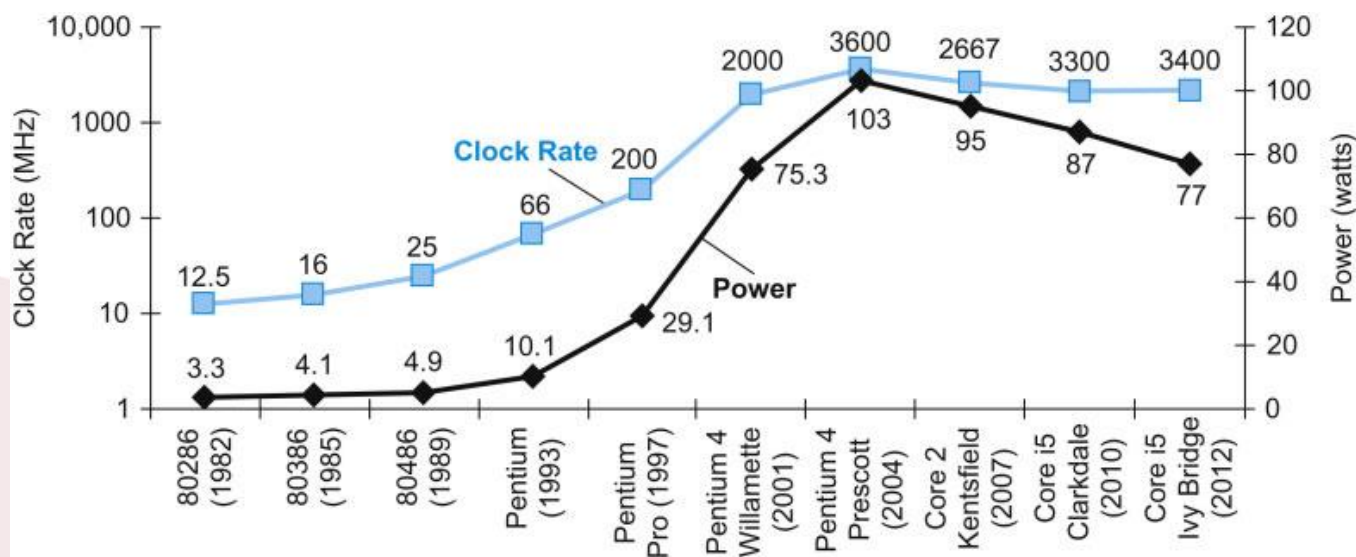


FIGURE 1.16 Clock rate and Power for Intel x86 microprocessors over eight generations and 25 years. The Pentium 4 made a dramatic jump in clock rate and power but less so in performance. The Prescott thermal problems led to the abandonment of the Pentium 4 line. The Core 2 line reverts to a simpler pipeline with lower clock rates and multiple processors per chip. The Core i5 pipelines follow in its footsteps.

# Conclusões



- Desempenho é específico de cada programa
  - Qualquer medida de desempenho deve refletir o tempo de execução
  - Tempo de execução total é um consistente resumo para desempenho
- Para um dado ISA, melhorias no desempenho são conseguidas com:
  - Aumento da frequência do clock (sem aumentar CPI)
  - Melhorias na organização do processador que diminua CPI
  - Melhorias no compilador que reduza CPI ou número de instruções
  - Escolhas do algoritmo ou da linguagem que afete número de instruções
- O que deve ser evitado ...
  - Usar subconjunto da equação de desempenho como métrica
  - Supor que a melhoria de um aspecto do computador vai ocasionar um melhoria no desempenho proporcional ao tamanho da melhoria

